50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. DCA/SW/MT-88/001e	2.	3. Recipient's Accession No

4. Title and Subtitle
Defense Communications Agency
Upper Level Protocol Test System
Internet Protocol Remote Driver Specification

5. Report Date
May 1988

6.

7. Author(s)

8. Performing Organization Rept. No

9. Performing Organization Name and Address
Defense Communications Agency
Defense Communications Engineering Center
Code R640
1860 Wiehle Ave.
Reston, VA 22090-5500

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.
(C)
(G)

12. Sponsoring Organization Name and Address

13. Type of Report & Period Covered
FINAL

14.

15. Supplementary Notes
For magnetic tape, see: AD-[illegible]128

16. Abstract (Limit: 200 words)

This document is part of a software package that provides the capability to conformance test the Department of Defense suite of upper level protocols including: Internet Protocol (IP) Mil-Std 1777, Transmission Control Protocol (TCP) Mil-Std 1778, File Transfer Protocol (FTP) Mil-Std 1780, Simple Mail Transfer Protocol (SMTP) Mil-Std 1781 and TELNET Protocol Mil-Std 1782.

Document Analysis a. Descriptors
Protocol Test Systems
Conformance Testing
Department of Defense Protocol Suite

b. Identifiers/Open-Ended Terms
Internet Protocol (IP)
Transmission Control Protocol (TCP)
File Transfer Protocol (FTP)
Simple Mail Transfer Protocol (SMTP)
TELNET Protocol

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Unlimited Release	UNCLASSIFIED	24
	20. Security Class (This Page) UNCLASSIFIED	22. Price

DEFENSE COMMUNICATIONS AGENCY

UPPER LEVEL PROTOCOL TEST SYSTEM

INTERNET PROTOCOL MIL-STD 1777
REMOTE DRIVER SPECIFICATION

MAY 1988

Disclaimer Concerning Warranty and Liability

This software product and documentation and all future updates to it are provided by the United States Government and the Defense Communications Agency (DCA) for the intended purpose of conducting conformance tests for the DoD suite of higher level protocols. DCA has performed a review and analysis of the product along with tests aimed at insuring the quality of the product, but does not warranty or make any claim as to the quality of this product. The product is provided "as is" without warranty of any kind, either expressed or implied. The user and any potential third parties accept the entire risk for the use, selection, quality, results, and performance of the product and updates. Should the product or updates prove to be defective, inadequate to perform the required tasks, or misrepresented, the resultant damage and any liability or expenses incurred as a result thereof must be borne by the user and/or any third parties involved, but not by the United States Government, including the Department of Commerce and/or The Defense Communications Agency and/or any of their employees or contractors.

Distribution and Copyright

Comments

Comments or questions about this software product and documentation can be addressed in writing to:

DCA Code R640
1360 Wiehle Ave
Reston, VA 22090-5500
ATTN: Protocol Test System Administrator

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1 - SCOPE AND PURPOSE

This specification describes the program functions of the IP Remote Driver, the software to exercise a vendor's Internet Protocol (IP) implementation (IUT) for MIL-STD-1777. Because it runs on the vendor's host rather than in the Protocol Test System, the software is called a Remote Driver (RD). An IP host/Remote Driver must also be supplied for gateway IP testing.

SECTION 2 - THE PROTOCOL TEST SYSTEM

2.1 DESIGN

The Protocol Test System (PTS) exercises an IUT performing peer protocol exchanges with a reference implementation at the Protocol Test System site. A driver controls each peer protocol through its upper level interface. To generate reproducible results, a script controls each driver.

The major components of the Protocol Test System are:

- Central Driver--to coordinate and monitor protocol testing;
- Remote Driver--to exercise the IP IUT
- Laboratory Slave Driver--to exercise the IP Reference implementation, record IP datagrams exchanged over the test connections, and provide communication links with the Central Driver.

The Central Driver, which coordinates and monitors protocol testing, analyzes the input it receives from the Laboratory Slave Driver to determine the success or failure of each test.

2.2 FUNCTIONS

The IP RD makes it possible for the Central Driver to interact with the IP IUT. The RD establishes a connection through its local IP and waits for datagrams from the Laboratory Slave Driver's (LSD) address. When such a datagram arrives, the RD parses the data area in a search for commands that instruct it to form a datagram to the LSD. If unable to respond as directed, the RD notifies the LSD by means of a mutually known set of messages that are described in Section 4.2. All data received from the IUT are forwarded to the Central Driver.

SECTION 3 - ENVIRONMENT

3.1 SUPPORT SOFTWARE

The Internet Protocol Remote Driver may reside on any computer that supports the Military Standard IP. In addition, the IP RD may be written in any language and may have any type of system support.

3.2 INTERFACES

3.2.1 Upper Layer Protocol to Internet Protocol.

Table 1 shows the minimum information that the IP RD must be able to send to the IP IUT (Internet Protocol Implementation Under Test).

PARAMETER	DEFINITION
Source Address	- Address to be placed in source field of IP packet header.
Destination Address	- Address to be placed in destination field of IP packet header.
Protocol	- Upper Layer Protocol Number.
Type of Service	- Values indicating: Precedence Delay Throughput Reliability.
Don't Fragment	- Indication to set the IP don't fragment option.
Time-to-Live	- Indication to set the IP packet's useful lifetime.
Length	- Indication of IP packet's desired length.
Option	- Option field requested to be placed in IP packet header
Data	- Data for IP to pack into data field.
Security level	- Security level for security option requested to be placed in IP header
Accreditation Authority	- Accreditation authority to be placed with preceding security level in IP header

TABLE 1. IP Parameters

3.2.2 Internet Protocol to Upper Layer Protocol.

Table 2 shows the minimum information from the IP IUT that the IP RD must be able to recognize.

PARAMETER	DEFINITION
Source Address	- Address from source field of IP packet header.
Destination Address	- Address from destination field of IP packet header.
Protocol	- Upper Layer Protocol Number.
Type of Service	- Values indicating: Precedence Delay Throughput Reliability.
Length	- Length of IP packet.
Data	- Data from IP data field.
Error	- Error indication from IP.

TABLE 2. IP Responses

SECTION 4 - DATAGRAM PRESENTATION

Each datagram sent to the IUT consists of a series of independent commands. Except for **ident**, a command may appear in any position: The **ident** command will always be first in the sequence. Figure 2 shows a typical datagram that will be sent to the IUT.

ident = 12345 proto = 255 prec = 7 ipttl = 25 datalen = 128

FIGURE 1. IP RD Command String

4.1 COMMAND COMPONENTS

Figure 3 shows the syntax of most IP Remote Driver commands, which consist of a keyword and a decimal value.

keyword<space>=<space>value<space>

FIGURE 2. IP RD Command Syntax

The only exceptions are the **end** and **option** commands. The **end** command consists of the "end" keyword by itself. The **option** command consists of the "option" keyword immediately followed by up to 40 bytes that occupy the remainder of the data field.

All bytes in the data field are ASCII characters except for the bytes that follow an "option" keyword.

4.1.1 Keywords.

Table 3 lists the keywords for Remote Drivers.

KEYWORDS	SEMANTICS
source	- Use source address that follows.*
dest	- Send message to address that follows.*
proto	- Use protocol that follows.
prec	- Use precedence that follows.
reliab	- Turn on reliability bit.
delay	- Turn on delay bit.
throu	- Turn on throughput bit.
ident	- Use identification that follows or any other value permitted by the Military Standard.
dontfrag	- Turn on don't fragment flag.
ipttl	- Use time-to-live that follows.
datalen	- Send back a message of the following length.
option	- Use the following option string to send this message.*
security	- Send back a message containing a security option set with this security level and the accreditation provided in a companion command.
accreditation	- Use this accreditation with the security level provided in a companion command.
end	- End test.

TABLE 3. Remote Driver Keywords

***Note: All addresses are given in network octal.**

4.1.2 Values.

Each keyword can be assigned a range of values. Table 4 lists each keyword, its corresponding range of values, and the value that should be set as the default.

KEYWORD	RANGE OF VALUES	DEFAULT
source	---	IUT's address
dest	---	LSD's address
proto	0 - 255	255
prec	0 - 7	0
reliab	0 - 1	0
delay	0 - 1	0
throu	0 - 1	0
ident	0 - 65535	1
dontfrag	0 - 1	0
ipttl	0 - 255	15
datalen	0 - 65535	length of received datagram's data field
security	0 - 65535	0
accreditation	0 - 65535	0

TABLE 4. Remote Driver Keywords and Values

4.2 RESPONSE COMPONENTS

Normally, the IP RD ensures that the header information is correctly set and that the data field is the requested length. The data field should be created by replicating or truncating the command string until the requested length is obtained.

Table 5 lists appropriate responses for two inoperative conditions. If the IUT does not implement one of the requested commands, it should place the **ident** command followed by the string "notimpl" in the response datagram's data field. If the IUT rejects the response datagram, it should send another datagram with the **ident** command followed by the string "error" in the datagram's data field.

RESPONSE	INSTRUCTIONS
error	- If the IP IUT does not accept a datagram, send back response to Central Test Driver.
notimpl	- If the IP IUT does not implement a command, send back response to Central Test Driver.

TABLE 5. Remote Driver Responses

SECTION 5 - DESIGN DETAILS

The following sections explain the initialization and operation of a Remote Driver.

5.1 CONNECTION ESTABLISHMENT

At runtime, the RD should perform any internal initialization necessary, and then wait at an address obtainable from the Protocol Test System testing facility for a connection from the Laboratory Slave Driver. This address is always an address on the same proximate net as the Protocol Test System testing facility.

5.2 COMMAND RECEPTION

Each RD must accept datagrams of up to 1024 bytes, with a protocol number of 255. Note the Central Test Authority can change the protocol number to something else if necessary. The RD must recognize all the keywords listed in Table 3, and may notify the Laboratory Slave Driver of any it does not implement. (See Section 4.2.)

For the accepted commands, the RD applies all implementation details required to ensure that the semantics of the commands are executed. This execution normally will involve creating an IP packet with the appropriate settings in the header and having the IP IUT send the packet to the LSD. The data portion of the IP packet should be the original command string reproduced as often as necessary to satisfy any packet length request in the original command string.

5.3 PACKET TRANSMISSION

After the RD has created the test IP packet according to the instructions of the command string, it attempts to send the packet back to the Laboratory Slave Driver via the IP IUT. If the IP IUT rejects the packet, the RD sends an error response packet instead. The IP header identification number of the error response packet should be set as indicated by the **ident** command. All other IP header fields should be set to their Table 4 default values. The data field should contain the **ident** command, followed by the string "error". (Note: Sending an error response packet does not necessarily constitute an error in the IP IUT.)

5.4 CONNECTION TERMINATION

Upon receipt of the command **end** (or upon a fatal error in the local implementation), the Remote Driver closes the IP connection with the Laboratory Slave Driver. No response datagram is expected or permitted.

APPENDIX A - References

"Military Standard Internet Protocol," MIL-STD-1777 (12 August 1983)

Internet Protocol Programmers and Users Manual, SDC TM-WD-8801/017/00

DCA Protocol Laboratory Functional Description, SDC TM-WD-8801/015/00

APPENDIX B - Glossary

This list of terms and acronyms does not include item names or data codes.

ASCII	- American Standard Code for Information Interchange. A data communications code set.
CD	- Central Driver
IP	- Internet Protocol
IP RD	- Internet Protocol Remote Driver
LSD	- Laboratory Slave Driver
RD	- Remote Driver
ULP	- Upper Layer Protocol
Datagram	- A self-contained package of data carrying enough information to be routed from source to destination without reliance on earlier exchanges between source or destination and the transporting subnetwork.
Keyword	- A lowercase ASCII string with a defined meaning.

APPENDIX C - Program Logic

```
[ IP Remote Driver                                                    ]
[                                                                     ]
[ Brian Griffin                                                       ]
[                                                                     ]
[ 9/15/86                                                             ]
[                                                                     ]
[ This PDL represents an Internet Protocol Remote Driver              ]
[  for MIL-STD-1777 certification testing. It assumes that            ]
[  the IP header is used to pass most of parameters for the           ]
[  section 6 interaction primitives.                                  ]
[                                                                     ]
[ MAJOR DATA STRUCTURES:                                              ]
[                                                                     ]
[  FROM_ULP        The response datagram molded by the Remote         ]
[                   Driver to be sent by the Implementation           ]
[                   Under Test.                                       ]
[                                                                     ]
[  TO_ULP          The datagram delivered by the IUT to the           ]
[                   Remote Driver and an error code.                  ]
[                                                                     ]
[ MINOR VARIABLES:                                                    ]
[                                                                     ]
[  DATA_LENGTH     The amount of data requested to be sent            ]
[                   in the response datagram.                         ]
[                                                                     ]
[  DATA_OFFSET     The offset within the datagram's data              ]
[                   field.                                            ]
[                                                                     ]
[  OPTION_OFFSET   The offset within the received datagram's          ]
[                   data field where 0 to 40 octets of the            ]
[                   requested response datagram's option              ]
[                   field can be found.                               ]
[                                                                     ]
[  TEST_ID         The value of the identification to be              ]
[                   used for the response datagram. It may            ]
[                   be needed to send a possible error or             ]
[                   not implemented response datagram.                ]
[                                                                     ]
[ NOTICE:                                                             ]
[                                                                     ]
[  This PDL may require some modification to work with the            ]
[   IP Implementation Under Test.                                     ]
```

```
BEGIN IP_REMOTE_DRIVER

[ Handle the MIL-STD-1777 section 6.2.2.1 DELIVER primitive.]

IF ( TO_ULP.PROTOCOL is 255) THEN

     [ Section 9.4.6.3.8, 9.4.6.3.10, or]
     [  9.4.6.3.12 datagram for Remote Driver]

     discard the TO_ULP.OPTION_DATA in the received datagram
      [using TO_ULP.INTERNET_HEADER_LENGTH]

     [ Store most IP Remote Driver Specification Table]
     [  4 default values in the response datagram.]

     set FROM_ULP.SOURCE_ADDR to the IUT's address

     set FROM_ULP.DESTINATION_ADDR to the CTD's address

     set FROM_ULP.PROTOCOL to 255

     set FROM_ULP.TYPE_OF_SERVICE to 0

     set FROM_ULP.DONT_FRAGMENT to 0

     set FROM_ULP.TIME_TO_LIVE to 15

     set FROM_ULP.DATA to TO_ULP.DATA


     set DATA_LENGTH to TO_ULP.LENGTH

     set OPTION_OFFSET to TO_ULP.LENGTH


     [ Parse the received datagram's data field and process
     [  the IP Remote Driver specification commands found.

     set DATA_OFFSET to 0 [offset of first data byte]

     FOR ( every IP Remote Driver command contained
             in the received datagram's data field
             [DATA_OFFSET less than TO_ULP.LENGTH] ) LOOP

          [ Look for next command in TO_ULP.DATA using
          [  DATA_OFFSET.
          [
          [ The Remote Driver need only look for the first
          [  few characters that uniquely identify the
          [  command because the IP Test System will not
          [  try to fool it.
```

```
IF ( next command is "ident") THEN

     add 8 to DATA_OFFSET to advance past "ident = "

     convert decimal ASCII value at DATA_OFFSET and
      store its value in FROM_ULP.IDENTIFIER and
      TEST_ID

     advance DATA_OFFSET past the space character that
      follows the decimal ASCII value

ELSEIF ( next command is "datalen") THEN

     add 10 to DATA_OFFSET to advance past "datalen = "

     convert decimal ASCII value at DATA_OFFSET and
      store its value in DATA_LENGTH

     advance DATA_OFFSET past the space character that
      follows the decimal ASCII value

     [ NOTICE: To permit the IP Test System to send     ]
     [  arbitrary length datagrams to the IUT, the      ]
     [  "datalen" command may be repeated more than     ]
     [  once and/or have a number of leading ASCII      ]
     [  zeros. The data length value represented is     ]
     [  guaranteed to be the same within any given      ]
     [  datagram.                                       ]

ELSEIF ( next command is "option") THEN

     add 6 to DATA_OFFSET to advance past "option"

     set OPTION_OFFSET to DATA_OFFSET to record the
      offset9 of the first byte which should be placed
      in the response datagram's FROM_ULP.OPTION_DATA

     set DATA_OFFSET to TO_ULP.LENGTH

ELSEIF ( next command is "prec") THEN

     add 7 to DATA_OFFSET to advance past "prec = "

     convert decimal ASCII value at DATA_OFFSET and store
      its value in FROM_ULP.TYPE_OF_SERVICE.PRECEDENCE

     advance DATA_OFFSET past the space character that
      follows the decimal ASCII value
```

```
ELSEIF ( next command is "reliab") THEN

     add 9 to DATA_OFFSET to advance past "reliab = "

     convert decimal ASCII value at DATA_OFFSET and store
      its value in FROM_ULP.TYPE_OF_SERVICE.RELIABILITY

     advance DATA_OFFSET past the space character that
      follows the decimal ASCII value

ELSEIF ( next command is "delay") THEN

     add 8 to DATA_OFFSET to advance past "delay = "

     convert decimal ASCII value at DATA_OFFSET and store
      its value in FROM_ULP.TYPE_OF_SERVICE.DELAY

     advance DATA_OFFSET past the space character that
      follows the decimal ASCII value

ELSEIF ( next command is "throu") THEN

     add 8 to DATA_OFFSET to advance past "throu = "

     convert decimal ASCII value at DATA_OFFSET and store
      its value in FROM_ULP.TYPE_OF_SERVICE.THROUGHPUT

     advance DATA_OFFSET past the space character that
      follows the decimal ASCII value

ELSEIF ( next command is "ipttl") THEN

     add 8 to DATA_OFFSET to advance past "ipttl = "

     convert decimal ASCII value at DATA_OFFSET and store
      its value in FROM_ULP.TIME_TO_LIVE

     advance DATA_OFFSET past the space character that
      follows the decimal ASCII value

ELSEIF ( next command is "proto") THEN

     add 8 to DATA_OFFSET to advance past "proto = "

     convert decimal ASCII value at DATA_OFFSET and store
      its value in FROM_ULP.PROTOCOL

     advance DATA_OFFSET past the space character that
      follows the decimal ASCII value
```

```
        ELSEIF ( next command is "dest") THEN

             add 7 to DATA_OFFSET to advance past "dest = "

             convert decimal ASCII value at DATA_OFFSET and store
              its value in FROM_ULP.DESTINATION_ADDR

             advance DATA_OFFSET past the space character that
              follows the decimal ASCII value

        ELSEIF ( next command is "source") THEN

             add 9 to DATA_OFFSET to advance past "source = "

             convert decimal ASCII value at DATA_OFFSET and store
              its value in FROM_ULP.SOURCE_ADDR

             advance DATA_OFFSET past the space character that
              follows the decimal ASCII value

        ELSEIF ( next command is "dontfrag") THEN

             add 11 to DATA_OFFSET to advance past "dontfrag = "

             convert decimal ASCII value at DATA_OFFSET and store
              its value in FROM_ULP.DONT_FRAGMENT

             advance DATA_OFFSET past the space character that
              follows the decimal ASCII value

        ELSE

             ["end" command]

             terminate IP Remote Driver execution

        ENDIF

   END_LOOP


   IF ( "option" command found
        [OPTION_OFFSET less than FROM_ULP.LENGTH]) THEN

        IF ( unable to expand datagram to hold option) THEN

             log the test identification number TEST_ID so that
              the cause of the test failure can be reported to
              the Protocol Laboratory Manager [optional]

             take implementation dependent action to make the
              IUT look as good as possible
```

```
     ELSE

          insert the bytes supplied by the "option" command
           into FROM_ULP.OPTION_DATA [using TO_ULP.DATA,
           OPTION_OFFSET and TO_ULP.LENGTH]

          indicate length of FROM_ULP.OPTION_DATA
           [set FROM_ULP.INTERNET_HEADER_LENGTH to 5 plus
           (the number of 4 octet units needed to store the
           option bytes supplied by the "option" command)]

     ENDIF
ENDIF


IF ( "datalen" command requires data replication
     [DATA_LENGTH greater than FROM_ULP.LENGTH]) THEN

     IF ( unable to expand datagram to hold data) THEN

          log the test identification number TEST_ID so that
           the cause of the test failure can be reported to
           the Protocol Laboratory Manager [optional]

          take implementation dependent action to make the
           IUT look as good as possible

     ELSE

          append copies of the received data field to
           FROM_ULP.DATA so that the length of FROM_ULP.DATA
           is DATA_LENGTH using only a leading portion for
           the last copy if DATA_LENGTH is not an integer
           multiple of the received datagram's data length

          [ FOR EXAMPLE:                                      ]
          [                                                   ]
          [ received data field:                              ]
          [                                                   ]
          [   "ident = 502 datalen = 42 "                     ]
          [                                                   ]
          [ response data field:                              ]
          [                                                   ]
          [   "ident = 502 datalen = 42 ident = 502 datal"    ]

     ENDIF

ELSEIF ( "datalen" command requires data truncation
         [DATA_LENGTH less than FROM_ULP.LENGTH]) THEN
```

```
        truncate the trailing portion of FROM_ULP.DATA so that
         its length equals DATA_LENGTH

        [ FOR EXAMPLE:                                                  ]
        [                                                               ]
        [ received data field:                                          ]
        [                                                               ]
        [   "ident = 503 datalen = 13 "                                 ]
        [                                                               ]
        [ response data field:                                          ]
        [                                                               ]
        [   "ident = 503 d"                                             ]

ENDIF

set FROM_ULP.LENGTH to its proper value [DATA_LENGTH]


Invoke the SEND service request primitive using FROM_ULP


IF ( TO_ULP.ERROR indicates that the request
     is illegal [according to section 9.4.6.2.9]) THEN

     IF ( able to obtain buffer space to send
          an IP Remote Driver error response packet) THEN

          [ Form an IP Remote Driver error response]
          [  packet using most Table 4 default values.]

          set FROM_ULP.SOURCE_ADDR to the IUT's address

          set FROM_ULP.DESTINATION_ADDR to the CTD's address

          set FROM_ULP.PROTOCOL to 255

          set FROM_ULP.TYPE_OF_SERVICE to 0

          set FROM_ULP.IDENTIFICATION to TEST_ID

          set FROM_ULP.DONT_FRAGMENT to 0

          set FROM_ULP.TIME_TO_LIVE to 15

          set FROM_ULP.DATA to the ident command
           reconstructed from TEST_ID followed
           by the string "error"

          set FROM_ULP.LENGTH to its proper value
           [length of ident command and "error" string]
```

```
            set FROM_ULP.OPTION_DATA
             [and FROM_ULP.INTERNET_HEADER_LENGTH] to
             request an empty option field


            Invoke the SEND service request primitive
             using FROM_ULP

      ENDIF

ELSEIF ( TO_ULP.ERROR indicates that the request is legally
         not implemented [excessive datagram length]) THEN

      IF ( able to obtain buffer space to send an IP
           Remote Driver not implemented response packet) THEN

            [ Form an IP Remote Driver not implemented response]
            [  packet using most Table 4 default values.       ]

            set FROM_ULP.SOURCE_ADDR to the IUT's address

            set FROM_ULP.DESTINATION_ADDR to the CTD's address

            set FROM_ULP.PROTOCOL to 255

            set FROM_ULP.TYPE_OF_SERVICE to 0

            set FROM_ULP.IDENTIFICATION to TEST_ID

            set FROM_ULP.DONT_FRAGMENT to 0

            set FROM_ULP.TIME_TO_LIVE to 15

            set FROM_ULP.DATA to the ident command
             reconstructed from TEST_ID followed
             by the string "notimpl"

            set FROM_ULP.LENGTH to its proper value
             [length of ident command and "notimpl" string]

            set FROM_ULP.OPTION_DATA
             [and FROM_ULP.INTERNET_HEADER_LENGTH] to
             request an empty option field


            Invoke the SEND service request primitive
             using FROM_ULP

      ENDIF

ENDIF
```

```
ELSE
     [ possible 9.4.6.3.1 ICMP message ]
     implementation dependent [discard the datagram]
ENDIF
END IP_REMOTE_DRIVER
```